

SNELL & WILMER L.L.P.  
Alan L. Sullivan (3152)  
Todd M. Shaughnessy (6651)  
Amy F. Sorenson (8947)  
15 West South Temple, Suite 1200  
Salt Lake City, Utah 84101-1004  
Telephone: (801) 257-1900  
Facsimile: (801) 257-1800

CRAVATH, SWAINE & MOORE LLP  
Evan R. Chesler (admitted pro hac vice)  
David R. Marriott (7572)  
Worldwide Plaza  
825 Eighth Avenue  
New York, New York 10019  
Telephone: (212) 474-1000  
Facsimile: (212) 474-3700

*Attorneys for Defendant/Counterclaim-Plaintiff  
International Business Machines Corporation*

**IN THE UNITED STATES DISTRICT COURT  
FOR THE DISTRICT OF UTAH**

THE SCO GROUP, INC.,

Plaintiff/Counterclaim-Defendant,

v.

INTERNATIONAL BUSINESS  
MACHINES CORPORATION,

Defendant/Counterclaim-Plaintiff.

**DECLARATION OF  
TODD M. SHAUGHNESSY**

Civil No. 2:03CV-0294 DAK

Honorable Dale A. Kimball  
Magistrate Judge Brooke C. Wells

RECEIVED CLERK  
**FILED**  
2005 MAY -3 P 7:02

U.S. DISTRICT COURT  
DISTRICT OF UTAH

I, Todd M. Shaughnessy, declare as follows:

1. I represent International Business Machines Corporation ("IBM") in the above-entitled action brought by The SCO Group, Inc. ("SCO"). This declaration is submitted pursuant to the Court's January 18, 2005 Order Concerning SCO's Renewed Motion to Compel (the "Order").

2. The Court ordered IBM to produce CMVC and RCS data relating to IBM's AIX and Dynix operating systems, including "all versions and changes to AIX and Dynix" (Order at 9-10), and to produce information regarding the 3,000 AIX and Dynix developers who "made the most contributions and changes to the development of AIX and Dynix". (Order at 16.) With respect to the source code produced from CMVC and RCS, the Court ordered IBM to submit an affidavit "specifying the efforts it took to deliver the code from the CMVC and RCS systems". (Order at 10.) With respect to information about the 3,000 AIX and Dynix programmers who "made the most contributions and changes to the development of AIX and Dynix" the Court ordered IBM to submit an affidavit "detailing the process by which the 3,000 were chosen". (Order at 17.)

3. As described in more detail below, IBM has complied with the Court's Order, and has produced all responsive, non-privileged information located after an extensive search. As ordered by the Court, IBM produced from CMVC and from RCS all source code relating to the AIX and Dynix operating systems, including all versions and changes to the code. IBM also produced from CMVC and RCS all documentation related to the AIX and Dynix operating systems, including all programmer's notes, design documents, and white papers. IBM identified all the individuals who created or made changes to AIX or Dynix source code, as recorded by CMVC and RCS, prepared a list of those individuals, together with their login identifiers and contact information (for every person for whom IBM had that information), and provided that

list to counsel for SCO on May 3, 2005. As explained below, the number of individuals who contributed source code to AIX and Dynix (as recorded by CMVC and RCS) is less than 3,000; therefore, the individuals identified for SCO constitute all of the individuals that are identified in CMVC and RCS as having made changes to AIX or Dynix. IBM has produced, in the form of CMVC and RCS data, information that shows what changes to the source code were specifically made by each of these individuals. As provided for by the Court in its April 20, 2005 Order Concerning IBM's Motion for Reconsideration, IBM has not searched for and through the files of 3,000 individuals. In accordance with that April 20 Order, IBM will produce, by July 19, 2005, documents from the files of the 100 individuals who made the most contributions and changes to AIX and Dynix source code.

4. IBM also undertook a reasonable search for programmer's notes, design documents, white papers and source code related to the AIX and Dynix operating systems that are not stored in CMVC or RCS and has completed its production of these documents to SCO.

5. Complying with the Court's Order involved more than 4,700 hours of work from more than 400 IBM employees. This does not include the time spent by IBM's counsel and consultants on this project, which was likewise considerable. IBM produced a total of more than 80 GB of source code and other electronic data to SCO, and more than 900,000 pages of paper (which were scanned and produced in electronic form on CDs).

6. Section I describes the steps IBM took to produce AIX source code, documentation (including programmer's notes, design documents, and white papers), and other information related to the AIX operating system from IBM's CMVC system. Section II describes the steps IBM took to produce Dynix source code, documentation (including programmer's notes, design documents, and white papers), and other information related to the Dynix operating system from IBM's RCS system. Section III describes the steps IBM took to

search for, collect, and produce AIX source code, programmer's notes, design documents, and white papers outside of IBM's CMVC system. Section IV describes the steps IBM took to search for, collect, and produce Dynix source code, programmer's notes, design documents, and white papers outside of IBM's RCS system. Section V describes IBM's production of information concerning each of the individuals who made changes to AIX or Dynix, including the names and contact information for these individuals, and what changes each individual specifically made.

**I. Production of AIX Code and Documents from CMVC**

7. CMVC is the source code revision system currently used by IBM's AIX development organization. CMVC has been used in AIX development since 1991. Other than the AIX source code stored in CMVC, IBM does not maintain revision control information for AIX prior to 1991. CMVC does not contain any source code or other information for the Dynix operating system.

8. CMVC provides shared access to source files used in the development of the AIX operating system, allows IBM to keep track of changes that are made to source code files, and ensures that the files are available for viewing or updating only by those with the proper authorization.

9. In accordance with the Court's January 18, 2005 Order, IBM identified and extracted from CMVC all of the source code, documentation, and other information related to the AIX operating system, built an AIX server loaded with the appropriate version of CMVC along with the source code and documentation related to the AIX operating system, tested the system to ensure it was functional, and delivered and installed the server to allow access by SCO.

10. The server contained a fully functional version of the CMVC tool, one hundred percent (100%) of the source code in CMVC that is part of or related to AIX (including the

operating system itself, development tools, documentation, and test programs) and one hundred percent (100%) of the documentation in CMVC that is related to AIX, including programmer's notes and design documents. One CMVC design document was redacted to protect attorney-client privileged information. After redaction, IBM was unable to restore the document into the database in electronic form. IBM produced the redacted version of the document along with the CMVC server. The code and documentation that IBM produced from CMVC represent more than 62 GB of data.

11. The particular CMVC server at IBM that contains source code and information related to AIX also contains a large amount of source code and material that is neither part of, nor related to, AIX. IBM did not produce source code or material in CMVC for components that are unrelated to AIX or its code, internal design, or methods. IBM excluded components containing design, manufacturing, and test information specific to IBM hardware products, such as hardware system designs, hardware test exercisers and other hardware test programs, and hardware manufacturing-related components. IBM also excluded firmware source code (machine-level code, distinct from the operating system, that is embedded into a computer hardware device or placed on a computer system to function at a level below the computer's operating system) and other software programs that are distinct from the operating system, such as middleware (software that provides support functions for software applications, such as application-to-application exchange of data, data storage management, and other services) and other applications.

12. The source code that is part of or related to the AIX operating system is not segregated in a single location within CMVC, but rather is commingled with hundreds of thousands of other source code files that are not part of or related to the AIX operating system. A thorough review of the contents of the CMVC system was undertaken to determine which of

the thousands of separate “components” within CMVC are part of or related to the AIX operating system.

13. A script—a small computer program—was written and executed to map each of the responsive components to the specific source code file names within CMVC. Using the list of file names and identifiers that had been generated, IBM then matched those file names and identifiers to corresponding Source Code Control System (“SCCS”) files. These SCCS files are the files maintained by IBM that provide the file development history since 1991 (or the inception of the file) for the particular corresponding source code file in the AIX operating system or related source code. These SCCS files were produced by IBM and allow SCO to reconstruct every version and iteration of AIX since 1991.

14. After all of the source code components for the AIX operating system were identified, the non-source code materials in CMVC that are related to the AIX operating system source code were similarly identified. This included programmer’s notes, design documents, and data about version control, users, and change histories.

15. CMVC programmer’s notes reflect developer commentary concerning defects and enhancements to AIX, and sometimes contain confidential information from IBM’s customers and vendors, or information covered by the attorney-client or work product privileges. If a CMVC programmer’s note contained third-party confidential information, the name of the third party (or other information that would identify the third party) was redacted from the copy of the programmer’s note to be produced to SCO. Reviewers also redacted privileged information from the copy of the note to be produced to SCO. All redacted information was marked with an appropriate legend. Out of 304,398 programmer’s notes produced from CMVC, approximately 100 contain a redaction of customer names or privileged information.

16. CMVC also contains more than 2,500 design documents related to AIX. These design documents were also produced to SCO. As noted above in paragraph 10, one design document was redacted to protect attorney-client privileged information and produced to SCO in redacted form.

17. For each source code file produced to SCO, IBM reviewed the origin codes or copyright notices in the code to identify potentially confidential third-party material. IBM located a copy of the relevant confidentiality terms and notified the third party prior to production, when required.

18. IBM obtained an AIX server with the hardware components necessary to produce the data from CMVC. An IBM team created a working copy of the CMVC source code revision system on the server. In order to retain CMVC database functionality that would allow SCO to search and query the code and documentation being produced, IBM copied the entire contents of the CMVC families that contained AIX-related content, and then removed the contents of the source files and programmer's notes that did not relate to AIX.

19. The server, which contained all the information described above, was made available to SCO at the offices of Snell & Wilmer in Salt Lake City, Utah on March 18, 2005. SCO's outside counsel took possession of this server. Along with the server, IBM also has made available to SCO general AIX and CMVC user documentation and a custom README file that contains basic instructions on how to start and navigate the server, CMVC, the necessary IDs and passwords, and a script to instruct SCO how to determine the changes made by each person who contributed code to AIX, as recorded by CMVC. A copy of the README file is attached to this Declaration as Exhibit A. A copy of the script is attached to this Declaration as Exhibit B.

## **II. Production of Dynix Code and Documents from RCS**

20. Revision Control System ("RCS") is the source code revision system that was used by Sequent's and IBM's Dynix development organization. It also serves as a shared electronic repository for programmer's notes, design documents, and white papers. The source code revision information in RCS dates back to 1988. Other than the Dynix source code stored in RCS, IBM has searched for, but has not been able to locate, revision control information for Dynix prior to 1988. RCS does not contain any source code or other information for the AIX operating system.

21. IBM has produced one hundred percent (100%) of the source code in RCS that is part of or related to Dynix (including the base operating system and layered products, development tools, and test programs). IBM also extracted, and produced to SCO, one hundred percent (100%) of the Dynix-related design documents, white papers, and programmer's notes that were stored in RCS.

22. The RCS server at IBM that contains source code and information related to Dynix also contains source code and material that is neither part of, nor related to, Dynix. IBM has not produced source code or material in RCS for components that are unrelated to Dynix or its code, internal design, or methods. IBM excluded components containing design, manufacturing, and test information specific to IBM or Sequent hardware products, such as hardware system designs, hardware test exercisers and other hardware test programs, and hardware manufacturing-related components. IBM also excluded firmware source code (machine-level code, distinct from the operating system, that is embedded into a computer hardware device or placed on a computer system to function at a level below the computer's operating system), and other software programs that are distinct from the operating system, such as middleware (software that provides support functions for software applications, such as

application-to-application exchange of data, data storage management, and other services) and applications.

23. Extracting the source code that is part of or related to the Dynix operating system required identification of the source code files that are not part of or related to the Dynix operating system. A thorough review of the contents of the RCS system was undertaken by IBM to determine which files are part of or related to the Dynix operating system.

24. Copies of both the source text file and the comma v file for each of the Dynix-related code files were extracted from RCS. Comma v files are the files maintained by RCS that provide the file development history since 1988 (or the inception of the file) for the particular corresponding source code file in the Dynix operating system or related source code. The copies were prepared in tape archive ("tar") format, and then compressed using a zip program to allow them to fit on the CDs. The total amount of this Dynix source code produced from RCS represents more than 17 GB of uncompressed data.

25. For each source code file produced to SCO, IBM reviewed the copyright notices in the code to identify potentially confidential third party material. IBM located a copy of the relevant confidentiality terms and notified the third party prior to production, when required.

### **III. Production of AIX Design Documents, Programmer's Notes, White Papers and Code Outside CMVC**

26. IBM also searched for design documents, programmer's notes, white papers and AIX source code that are not stored in the CMVC database and has completed its production of these documents. Certain AIX development teams keep a large portion of their work files and documents, other than what is required to be stored in CMVC, in shared electronic repositories. To collect a large volume of AIX design documents, programmer's notes, whitepapers, and code, and to avoid redundancy, IBM collected potentially responsive documents from shared electronic repositories at a department, team, and project level. These documents were reviewed for

responsiveness, third-party confidential information, and attorney-client privileged communications, and responsive, non-privileged documents have been produced to SCO.

27. IBM also located, from shared electronic repositories and from some data tapes, some source code for the AIX operating system. Although it is likely that this code is duplicative of the AIX source code already produced to SCO on the CMVC server as discussed in Paragraphs 7-19, IBM was unable to confirm that the code is duplicative, and therefore has produced this AIX source code to SCO, on CDs.

28. As I have noted above, IBM does not maintain revision control information for AIX source code pre-dating 1991. To the extent that any code for the AIX operating system (that did not duplicate the code already being produced in CMVC) was found during the search described in Paragraphs 26-27 above, it was produced. Paragraphs 29-31 below describe additional search efforts IBM undertook to locate pre-1991 versions of AIX code. No versions of AIX pre-dating 1991 were found.

29. In the 1980s and early 1990s, IBM prepared vital records backups of AIX source code and transferred them to a remote storage location. At some point in the 1990s, the AIX vital records tapes were transferred to Austin, Texas. In late 2000, the tapes were determined to be obsolete, and were not retained.

30. The AIX development organization contacted other IBM employees who were known or believed to have been involved with the development or product release of AIX versions prior to 1991. In addition, IBM managers and attorneys asked current members of the AIX development organization whether they were aware of the location of pre-1991 releases of AIX source code. No one asked was aware of any remaining copies of pre-1991 AIX source code.

31. Source code archives retained by the IBM group responsible for filing IBM copyright registrations and maintaining some of the IBM copyright records were transferred to IBM's Austin site in 2000. IBM searched those archives; all of the source code in the archives are duplicative of AIX versions and changes already produced on the CMVC server as discussed in Paragraphs 7-19.

**IV. Production of Dynix Design Documents, Programmer's Notes, White Papers and Code Outside RCS**

32. RCS is the shared electronic repository that was used by Dynix developers to store design documents, programmer's notes, and white papers. As discussed above, IBM collected responsive code and documents from RCS. In addition, IBM searched for and retrieved potentially responsive materials from archived Sequent records. These documents were reviewed for responsiveness, third-party confidential information, and attorney-client privileged communications, and all responsive, non-privileged documents have been produced to SCO.

33. As noted above, IBM searched for, but was unable to locate, revision control information for Dynix prior to 1988. IBM did locate some pre-1988 copies of archived Dynix source code files (without revision control information), which were produced to SCO on CDs.

**V. Contributors to AIX and Dynix**

34. As IBM previously noted in response to SCO's Interrogatory 5, the list of 7,200 individuals who have or have had access to AIX or Dynix source code are the people who work or worked on developing AIX and Dynix. Not all of these individuals, however, have made contributions or changes to AIX or Dynix source code; for example, a development supervisor may have access to CMVC or RCS, but may have never personally made any changes to the code. In response to the Court's order that IBM provide information as to which persons made contributions or changes to AIX or Dynix source code, IBM has identified the names, user IDs,

and contact information (to the extent IBM has such information in its records) for all of the individuals recorded by CMVC and RCS as having created or made changes to AIX or Dynix or related source code files, and has produced all such information to SCO.

35. The total number of individuals who are recorded by CMVC or RCS as having made contributions or changes to AIX or Dynix or related source code files is 2,704. This number, while less than the 3,000 individuals contemplated by the Order, includes all individuals who are recorded by CMVC and RCS as having made contributions and changes to AIX or Dynix.

36. The list of AIX contributors contains 2,234 names. These names were obtained by using CMVC tools to determine which CMVC users have ever created or modified AIX or related source code since CMVC versioning was initiated in 1991. This list includes all of the persons who are recorded by CMVC as having made changes to AIX source code. The list was examined manually to merge the data for users who had multiple IDs or name changes.

37. IBM has also produced to SCO the user IDs for all of the individuals who made changes or contributions to Dynix, as recorded by RCS. The list contains 470 user IDs and identifies the number of files created or modified by each user ID. IBM reconstructed and reviewed archived Sequent records and questioned former Dynix developers, and has provided to SCO all of the corresponding employee names and contact information that were obtained.

38. The CMVC and RCS revision control data produced by IBM include complete information (to the extent such information is recorded by CMVC or RCS) as to which individuals made which specific contributions or changes to AIX or Dynix source code, as well as when each such change was made.

39. For AIX, the contributions and changes made by each person can be determined by running a simple script, a copy of which was produced to SCO along with the CMVC system

on March 18, 2005. A copy of the script is also attached to this Declaration as Exhibit B. Using the script, SCO can type in any individual user ID, and the script will produce as its output a detailed list of all of the contributions and changes made by that user.

40. For Dynix, information about each change made to each file in the Dynix source code, including the date and time the change was checked-in to the RCS system, who checked-in the change, the number of lines of code added and deleted from the previous revision of the file, and a log message entered by the person who checked-in the change can be ascertained using standard RCS tools, such as the "rlog" command. For example, to determine the change history of the base\_callback.c,v file in the 4.6.1 version of the Dynix base operating system, SCO can type "rlog base\_callback.c,v", which results in the following output:

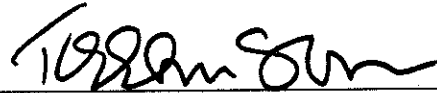
```
$ rlog base_callback.c,v

RCS file:      base_callback.c,v;   Working file:   base_callback.c
head:         1.4
branch:
locks:        ; strict
access list:
symbolic names: v4_6_lp: 1.4.3; v4_6_1: 1.4; v4_6_0p: 1.4.2; v4_6_0: 1.4;
comment leader: " * "
total revisions: 6;   selected revisions: 6
description:
base_callback.c
-----
revision 1.4
date: 97/09/29 18:20:23; author: mjs; state: Exp; lines added/del: 7/9
branches: 1.4.2; 1.4.3;
Made appropriate use of SYMUSED lint directive in this file.
PR #230499 / SCN rto1031.
-----
revision 1.3
date: 95/11/03 03:08:44; author: mjs; state: Exp; lines added/del: 5/2
lint fix.
-----
revision 1.2
date: 95/11/03 02:01:20; author: mjs; state: Exp; lines added/del: 20/2
Added lint ref for base_callback.
-----
revision 1.1
date: 95/11/02 20:14:52; author: mcneil; state: Exp;
Initial revision
-----
revision 1.4.3.1
date: 20/1./3. 6.:0.:6.; author: hbeare; state: Exp; lines added/del: 6/2
Branch for v4_6_lp
-----
revision 1.4.2.1
date: 20/0./9. 5.:8.:1.; author: breazile; state: Exp; lines added/del: 6/2
Branch for v4_6_0p
=====
```

41. I declare under penalty of perjury that the foregoing is true and correct.

Executed: May 3, 2005

Salt Lake City, Utah

A handwritten signature in black ink, appearing to read "Todd M. Shaughnessy", written over a horizontal line.

Todd M. Shaughnessy

**CERTIFICATE OF SERVICE**

I hereby certify that on the 3<sup>rd</sup> day of May, 2005, a true and correct copy of the foregoing was sent by U.S. Mail, postage prepaid, to the following:

Brent O. Hatch  
Mark F. James  
HATCH, JAMES & DODGE, P.C.  
10 West Broadway, Suite 400  
Salt Lake City, Utah 84101

Stephen N. Zack  
Mark J. Heise  
BOIES, SCHILLER & FLEXNER LLP  
100 Southeast Second Street, Suite 2800  
Miami, Florida 33131

Robert Silver  
BOIES, SCHILLER & FLEXNER LLP  
333 Main Street  
Armonk, New York 10504



---

Todd M. Shaughnessy

A

System operation (Please read the AIX 5.2 documentation before attempting to use this system)

- To power up the system, press the power button on the front panel.
- To reboot the system, run the 'reboot' command as root.
- To shutdown the system run the 'shutdown -F' command as root.
- To power up after shutdown, press the power button on the front panel.

CMVC operation (Please read the CMVC documentation before attempting to use the CMVC GUI or CMVC command line interfaces)

The aix and admin CMVC families will automatically start on system startup or reboot.

To start or restart the CMVC families, run the following commands as root:

- StartAIX.ksh - to restart the aix CMVC family
- StartAdmin.ksh - to restart the admin CMVC family

To run CMVC commands on the aix and admin CMVC families

- Log on the system as the scoid ID
- The scoid CMVC ID has hostlist access setup, no password needed to access the CMVC families
- The scoid CMVC ID has superuser authority to the families (refer to the CMVC docs for superuser description)
- To run the CMVC GUI interface run the 'cmvc' command (refer to the CMVC docs for help on the CMVC GUI)

To set the family to access from the CMVC GUI:

- Select "Set Family..." from the "Options" pull down
- Enter either "aix" or "admin" in the "Family:" field the < OK >

- or -

The "CMVC\_FAMILY" environment variable can be set to "aix" or "admin"

CMVC command line commands can be run from the command prompt (refer to the CMVC docs for help on the CMVC command line interface)

To set the family to access for the CMVC command line interface:

- The "-family" parameter on the command can be set to "aix" or "admin"

- or - - or -

The "CMVC\_FAMILY" environment variable can be set to "aix" or "admin"

aix family address: aix@<hostname>@7260

admin family address: admin@<hostname>@7265

user2diffs usage (user2diffs command is located in /home/scoid)

Log on as scoid

> user2diffs -?

Outputs the CMVC file changes for the specified CMVC login.

Usage: user2diffs [-E|-S -f family -v] login

-E Extract and diff rather than using sccsdiff

-S Use sccsdiff to get the file diffs

-f CMVC family (default is aix)

-v Status message level

For example,

/user2diffs leedp

will output all of the diffs for user 'leedp' in the 'aix' family.

/user2diffs -f admin arb

will output all of the diffs for user 'arb' in the 'admin' family'.

The output is sent to stdout.

Documentation (all docs are in /home/scoid; run "xpdf filename.pdf" to view PDF docs)

scoreadme.txt	Basic system information
aix52.pdf	AIX 5L Version 5.2: System User's Guide: Operating System and Devices
whatis.pdf	CMVC95: PDF for What is... files
howto.pdf	CMVC95: PDF for How to... files
gcommandinfo.pdf	CMVC95: PDF for General Command Information files
usersref.pdf	CMVC95: PDF for User's Reference files

System User ID's and passwords

ID	password	description
root	sc0root1	system admin ID
scoid	sc0cmvc1	cmvc ID



```
#!/bin/ksh
#####
# user2diffs
#
# ORIGINS: 27
#
# (C) COPYRIGHT International Business Machines Corp. 2005
# All Rights Reserved
# Licensed Materials - Property of IBM
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# Outputs the CMVC file changes for the specified user login.
#####
# @(#)0.11 user2diffs.sh 03/12/05 08:30:00

# Constants
#
integer Extract=0 ShellDebug=0 Verbose=0
typeset Prog=${0##*/}
typeset TmpDir=/tmp/$Prog.$$
export CMVC_FAMILY=aix

# Global variables
#
typeset Flag VcPath

# Write a message to stdout if $1 >= Verbose
#
function vStdout {
    integer MsgLevel

    MsgLevel=$1
    (( Verbose >= MsgLevel )) && {
        shift
        echo "$*"
    }
}

function vStderr {
    vStdout "$@" >&2
}

# Clean and exit
#
function Exit {
    typeset rc

    # Output a message if specified
    #
    rc=${1:-0}
    (( $# > 1 )) && {
        shift
        vStderr 0 "$*"
    }
}
```

```

        exit $rc
    }

    # Return the vc path for a given sourceid
    #
    function Sourceid2vc {
        (( ShellDebug )) && set -x

        echo $1 | awk -v vc=$VcPath '{
            file = "s."substr($0, length($0) - 1)
            t = substr($0, 1, length($0) - 2)
            if (length(t) < 4) t = "0"t
            while (length(t)) {
                if (path == "") {
                    if (length(t) > 4) {
                        path = substr(t, 1, 2)
                        t = substr(t, 3)
                    }
                    else {
                        path = substr(t, 1, 1)
                        t = substr(t, 2)
                    }
                }
                else {
                    path = path"/"substr(t, 1, 1)
                    t = substr(t, 2)
                }
            }
            print vc"/"path"/"file
            #printf("%s%s/%s\n", vc, path, file)
        }'
    }

    # Return the previous versionsid for the specified
    # pathname,releasename,versionsid
    #
    function PreviousVersionsid {
        Report -g "fileview f,versions v,versions x" \
        -w "f.nupathname = '$file' and f.releasename = '$rel' and v.sid = '$vers' and f.sourceid =
        v.sourceid and v.previousid = x.id" -sel x.sid
    }

    # Show the diffs between the specified versionsid and previous versionsid
    # using sccsdiff.
    # $1 = pathname
    # $2 = sourceid
    # $3 = versionsid
    # $4 = previous versionsid
    #
    function DiffBySccs {
        (( ShellDebug )) && set -x

        SccsFile=$(Sourceid2vc $2)
        if [[ -f $SccsFile ]]; then
            echo "====> $1 $4 -> $3"
        fi
    }

```

```

        sccsdiff -r$4 -r$3 $SccsFile
    else
        echo "====> $1 ($SccsFile) not found"
    fi
}

# Show the diffs between the specified pathname, releasename, versionsid
# and the previous versionsid
#
function DiffByExtract {
    (( ShellDebug )) && set -x

    Outfile1=$TmpDir/${1##*/}.$3
    Outfile2=$TmpDir/${1##*/}.$4
    echo "====> $1 $4 -> $3"
    mkdir -p $TmpDir
    File -extract $1 -rele $2 -vers $3 -stdout >$Outfile1
    File -extract $1 -rele $2 -vers $4 -stdout >$Outfile2
    diff $Outfile1 $Outfile2
    rm -rf $TmpDir
}

# Usage and exit
#
function Usage {
    exec >&2
    (( $# )) && echo "$*"
    cat <<-!

    Outputs the CMVC file changes for the specified CMVC login.

    Usage: $Prog [-E|-S -f family -v] login
    -E Extract and diff rather than using sccsdiff
    -S Use sccsdiff to get the file diffs
    -f CMVC family (default is aix)
    -v Status message level
    !
    exit 99
}

#####
# MAIN #
#####

# Parse the flags
#
while getopts ":ESXf:v" Flag; do
    case $Flag in
        E)
            Extract=1
            ;;
        S)
            Extract=0
            ;;
    esac
done

```

```

X)
    ShellDebug=1
    set -x
    ;;

f)
    export CMVC_FAMILY=$OPTARG
    ;;

v)
    Verbose=Verbose+1
    ;;

:)
    Usage "-$OPTARG requires an argument"
    ;;

?)
    [[ "$OPTARG" = "?" ]] && Usage
    Usage "illegal option -$OPTARG"
    ;;

esac

done
shift OPTIND-1

(( $# )) || Usage

# Set variables based on options
#
VcPath=/family/${CMVC_FAMILY%%@*}/vc
(( Extract == 0 )) && [[ ! -d $VcPath ]] &&
    Usage "Path to the SCCS files ($VcPath) does not exist.\nPerhaps you should use the -E
flag?"

#####
# Main #
#####
Report -g userview -w "login = '$1'" -sel id | read uid ;
Report -g changeview -w "userid = $uid" -sel pathname,releasename,versionsid |
while IFS='|' read file rel vers; do
    Report -g fileview -w "pathname = '$file' and releasename = '$rel'" \
        -sel sourceid,type | IFS='|' read sid ftype
    [[ "$ftype" = "binary" ]] && {
        echo "====> $file is type binary - skipping"
        continue
    }
    overs=$(PreviousVersionsid)
    if [[ -z "$overs" ]]; then
        echo "====> $file $vers is the original version"
    else
        if (( Extract )); then
            DiffByExtract $file $rel $vers $overs
        else
            DiffBySccs $file $sid $vers $overs
        fi
    fi
done

```

done fi